Quant Test Notebook

Before getting into details we would want to substantiate why we want rolling windows or as statistics calls them moving averages. This method addresses volitilty and removes cyclic patterns. It also allows a clear look at trends and long term patterns as well as changes in the structure, sudden shifts called regime change, without getting hung up on the well understoood structures of noise and smaller cycles. These smaller cycles and noise want to added back in later for projection or forecast. Given the assignment is windowing and design matrices for input into an algorithm I would argue using ARIMA (Auto Regressive Integrated Moving Average) a time series tool that has those windows already built in as well and the design matrix and model forecasting is a good way to start.

We aim for our model to be today is tomorrow plus trend plus noise at the time for each N y[t] = a[1]y[t-1] + Trend + Noise[t] for each time series N we want to forecast and utilizing the work of Box-Jenkins we want to start with an ARIMA (1,1,1) without a constant, formulated as

y[t] = Y[t]-Y[t-1] y[t] = phi[1]y[t-1] + e[t] - theta[1]e[t] where e[t] is the random noise also called shock, happening at time t; phi[t] is the AR[1] coeffient; theta[1] is the MA coeffient.

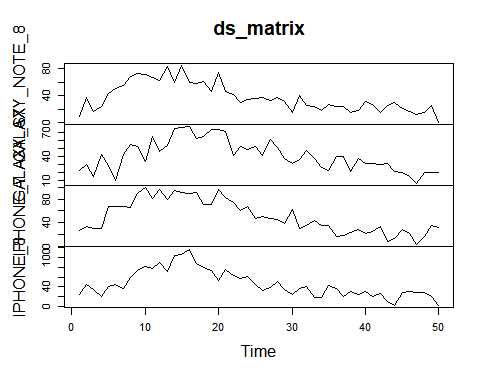
#some libs and data  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

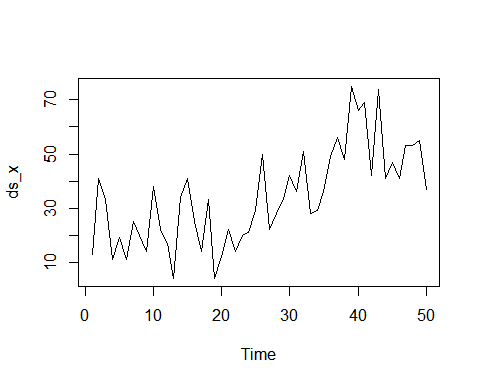
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(forecast)  
ds\_test\_data <- read.csv("~/APSQuant/ds\_test\_data.csv", header=TRUE)  
ds\_test<- select(ds\_test\_data, c(2,3,5,6))  
#could select training and testing sets by using window(ds\_test, start = 1, end = 20) or #similar  
#make timeseries object and check it  
ds\_x <- ts(ds\_test\_data$GALAXY\_S8, start = 1,frequency = 1)  
  
ds\_matrix <- ts(ds\_test, start = 1,frequency = 1)  
  
#head(ds\_matrix)  
  
plot.ts(ds\_matrix)



From the above we can describe the data as most peaking with a regime change at 15 and a decreasing tail. The galaxy S8 timeserries differs with a slow build and late peak almost as if it had had its time reversed from the rest of the set.

plot.ts(ds\_x)

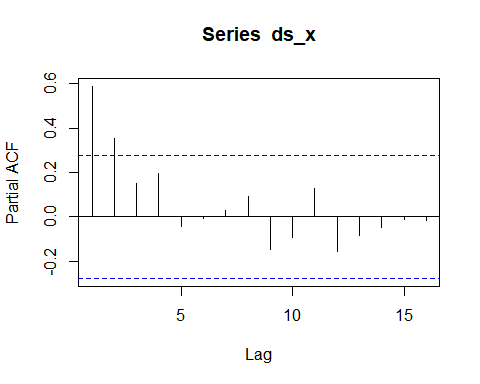


Then for each N one would plot autocorrelation and partial autocorrelation; auto tune and test an ARIMA model for n in N.

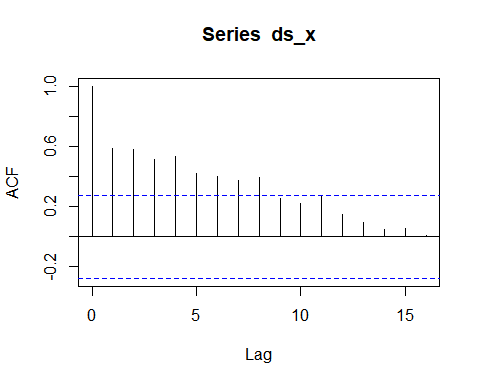
ds\_gs8 = auto.arima(ds\_x)  
summary(ds\_gs8)

## Series: ds\_x   
## ARIMA(0,1,1)   
##   
## Coefficients:  
## ma1  
## -0.6865  
## s.e. 0.1039  
##   
## sigma^2 estimated as 172.4: log likelihood=-195.51  
## AIC=395.01 AICc=395.27 BIC=398.8  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.744669 12.86347 10.47424 -23.23494 51.14063 0.8095233  
## ACF1  
## Training set -0.02517184

pacf(ds\_x)



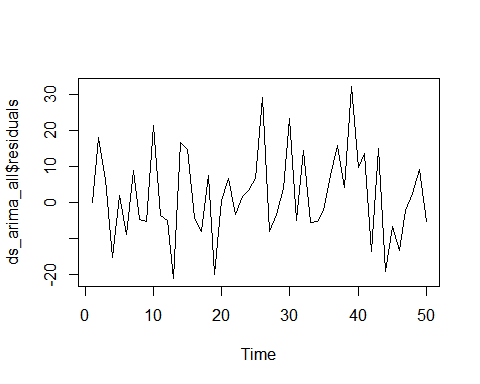
acf(ds\_x)



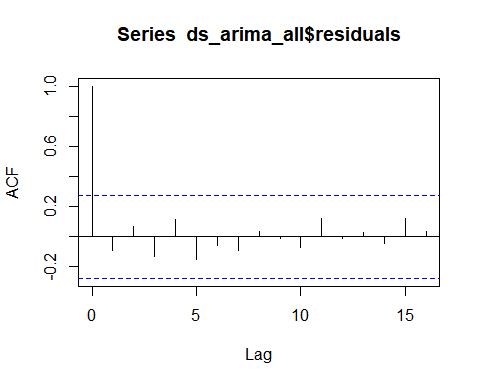
ds\_arima\_all <- auto.arima(ds\_x, xreg = ds\_matrix)  
  
summary(ds\_arima\_all)

## Series: ds\_x   
## Regression with ARIMA(0,1,1) errors   
##   
## Coefficients:  
## ma1 GALAXY\_NOTE\_8 GALAXY\_S7 IPHONE\_7 IPHONE\_8  
## -0.7286 0.1456 -0.0181 -0.3086 0.179  
## s.e. 0.1162 0.1636 0.1608 0.1615 0.133  
##   
## sigma^2 estimated as 170.4: log likelihood=-193.16  
## AIC=398.31 AICc=400.31 BIC=409.66  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 2.081663 12.24606 9.712395 -20.70924 47.89853 0.7506425  
## ACF1  
## Training set -0.09077132

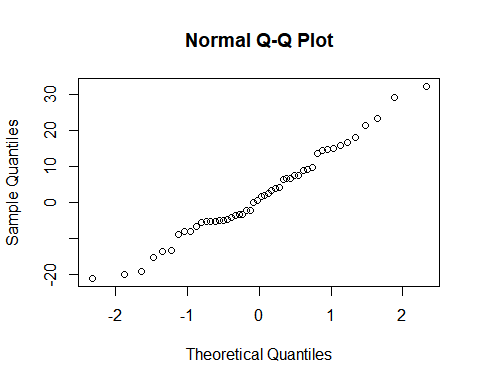
plot.ts(ds\_arima\_all$residuals)



acf(ds\_arima\_all$residuals)



qqnorm(ds\_arima\_all$residuals)



Here if we were actually forecasting this model we’d be doing phone\_ahead = forecast(ds\_x, h=5) plot.forecast(phone\_ahead) However, Let me stop here and point out that there are many models within time series analysis that could be quite interesting and a better fit for the work proposed. In particular VAR and VARIMA for multivariate modelling with timeseries. These ares introduce with the possibility of IRF(interoduced shock or noise) to test covariance.

Once one had decided on a best fit and method for each n in N you can train and test each model.

Add